

# Reducing Repair Traffic for Erasure Coding-Based Storage via Interference Alignment

Yunnan Wu  
Microsoft Research

Alexandros G. Dimakis  
California Institute of Technology

**Abstract**—We consider the problem of recovering from a single node failure in a storage system based on an  $(n, k)$  MDS code. In such a scenario, a straightforward solution is to perform a complete decoding, even though the data to be recovered only amount to  $1/k$ th of the entire data. This paper presents techniques that can reduce the network traffic incurred. The techniques perform algebraic alignment so that the effective dimension of unwanted information is reduced.

## I. INTRODUCTION

It is well known that erasure coding can be used in storage systems to efficiently store data while protecting against failures. For instance, we can divide a file of size  $B$  into  $k$  pieces, each of size  $B/k$ , encode them into  $n$  coded pieces using an  $(n, k)$  maximum distance separable (MDS) code, and store them at  $n$  nodes. Then, the original file can be recovered from any set of  $k$  coded pieces. In distributed storage systems based on  $(n, k)$  MDS codes, we are often faced with the *repair problem* [2]: if a node storing a coded piece fails or leaves the system, in order to maintain the same level of reliability, we need to create a new encoded piece and store it at a new node, but we can only access other encoded blocks. One straightforward way to do so is to let the new node download  $k$  encoded pieces from a subset of the surviving nodes, reconstruct the original file, and compute the needed new coded piece. In this process, the new node incurred a network traffic of  $k \times B/k = B$ . Since network bandwidth could be a critical resource in distributed storage systems, an important consideration is to conserve the repair network bandwidth. In our recent work [2], [5], we showed that it is possible to reduce this repair bandwidth below  $B$  and developed information theoretic lower bounds and achievable schemes. Note that in the problem setup of [2], [5] we allow the newly generated piece to be different from the failed piece, as long as the updated code maintains the same property (e.g., remains an  $(n, k)$ -MDS code). In essence, we were considering a *functional repair* problem. In this paper, we consider the problem of *exact repair*, i.e., we require that the failed piece is exactly reconstructed.

We start with an example. Consider a  $(4, 2)$ -MDS code defined over  $GF(5)$ , which is illustrated by the top part of Figure 1. There are four original information blocks,  $A_1$ ,  $A_2$ ,  $B_1$ , and  $B_2$ , of equal size. There are four storage nodes, shown in 4 rectangle boxes; each storage node stores two blocks. For example, node 1 stores  $A_1$  and  $A_2$ . It is easy to verify that this code can protect against any 2 node failures.

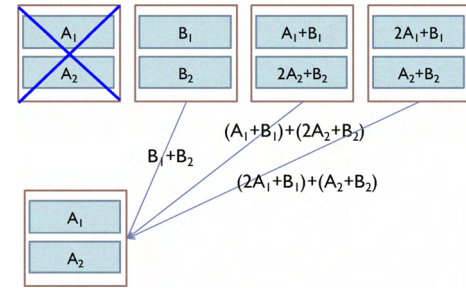


Fig. 1. Repairing a  $(4, 2)$ -MDS code, when node 1 fails.

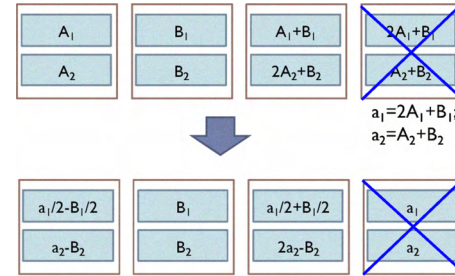


Fig. 2. Repairing the  $(4, 2)$ -MDS code when a non-systematic node fails.

Suppose the first node fails. To repair the failed node, the conventional method downloads 4 blocks from other nodes and then solves for the missing blocks  $A_1$  and  $A_2$ . Now we show a way to achieve the recovery while downloading only 3 blocks over the network. As illustrated by Figure 1, we download  $B_1 + B_2$  from the second node,  $(A_1 + B_1) + (2A_2 + B_2)$  from the third node (node 3 computes the sum of its two blocks and sends it over the network), and  $(2A_1 + B_1) + (A_2 + B_2)$  from the fourth node. Altogether, we have three equations and four unknowns. However, note that we can cancel out the term  $B_1 + B_2$  from  $(A_1 + B_1) + (2A_2 + B_2)$  and  $(2A_1 + B_1) + (A_2 + B_2)$ . Then we are left with two linearly independent equations involving  $A_1$  and  $A_2$ , which we can use to solve for  $A_1$  and  $A_2$ .

Similarly, if the second node fails, we can download 3 blocks over the network to recover it. What if a non-systematic node fails? We can perform a change of variables, as illustrated by Figure 2. The resulting situation is analogous to Figure 1.

The gist of Figure 1 is that when recovering  $A_1$  and  $A_2$ , all other nodes generate linear combinations of the form  $[A_1, A_2, B_1 + B_2]v$ . Therefore, the decoder effectively sees only 3 unknown blocks, even though there are 4 original unknowns. Since the decoder is only interested in recovering

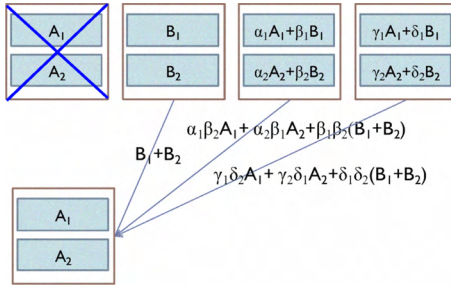


Fig. 3. A (4, 2)-MDS code with general coefficients.

$A_1$  and  $A_2$ ,  $B_1$  and  $B_2$  are essentially “interference signals”. In Figure 1, these interference signals are aligned along the same direction  $B_1 + B_2$  so that effectively their dimension is reduced from 2 to 1. This translates to savings in terms of the network bandwidth required for performing the code repair (partial recovery).

This technique has some similarity with the interference alignment technique for interference channels (see, e.g., [1] and the references therein), which we learned after obtaining the results of this paper. In communications, interference alignment refers to a scheme where the signals of multiple transmitters are carefully designed so that some signals cast “overlapping shadows” at the receivers that do not want them, while still enabling each receiver to distinguish its desired signal from others. To highlight the nature of the approach, we also use the term *interference alignment* to describe the schemes of this paper. However, there are significant differences between the interference alignment technique for communications and the interference alignment technique for storage presented herein (e.g., complex field vs. finite field, different topologies, and different alignment strategies).

## II. THE BASIC INTERFERENCE ALIGNMENT SCHEME

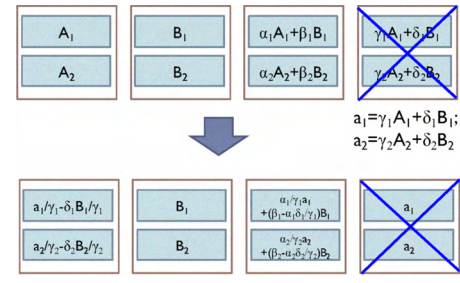
In this section, we show how to generalize the basic interference alignment technique to  $(n, k)$ -MDS codes. We begin by consider a concrete setup –  $(4, 2)$ -MDS, where we now allow general coefficients in the code. Then we discuss the general  $(n, k)$ -MDS case.

### A. (4, 2)-MDS with General Coefficients

First, we generalize Figure 1 by assuming some general coefficients in the code. We assume that node 3 stores  $\alpha_1 A_1 + \beta_1 B_1$  and  $\alpha_2 A_2 + \beta_2 B_2$  and node 4 stores  $\gamma_1 A_1 + \delta_1 B_1$  and  $\gamma_2 A_2 + \delta_2 B_2$ . Here  $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2, \delta_1, \delta_2$  are 8 coefficients; collectively, let  $\xi$  denote these 8 coefficients. Suppose the code is defined over a certain finite field  $\mathbb{F}$ . We shall examine the conditions on these coefficients for the interference alignment technique to be applicable.

In Figure 3, each row is an  $(4, 2)$ -MDS code involving two original information blocks  $A_i$  and  $B_i$ . The MDS condition requires that the two generator matrices satisfy the property that any  $k$  rows are linearly independent. Thus the MDS condition can be stated as requiring the product of 12 determinants being nonzero. For this example, the condition boils down to:

$$\alpha_1 \beta_1 \gamma_1 \delta_1 (\alpha_1 \delta_1 - \gamma_1 \beta_1) \alpha_2 \beta_2 \gamma_2 \delta_2 (\alpha_2 \delta_2 - \gamma_2 \beta_2) \neq 0. \quad (1)$$

Fig. 4. Repairing the  $(4, 2)$ -MDS code when a non-systematic part fails.

The left hand side of (1) is seen to be a multivariate polynomial in  $\xi$ ; denote this by  $P(\xi)$ .

Suppose node 1 fails. In recovering  $A_1, A_2$ , all nodes generate certain linear combinations so that the resulting blocks are functions of  $A_1, A_2$ , and  $B_1 + B_2$ . Thus, the interference signals  $B_1, B_2$  are aligned into one direction,  $B_1 + B_2$ . Specifically, node 3 will multiply  $\beta_2$  to its first block  $\alpha_1 A_1 + \beta_1 B_1$  and  $\beta_1$  to the second block  $\alpha_2 A_2 + \beta_2 B_2$ , and add them up to form  $\alpha_1 \beta_2 A_1 + \alpha_2 \beta_1 A_2 + \beta_1 \beta_2 (B_1 + B_2)$ . Similarly, node 4 would also generate a linear combination that aligns the interference subspace to  $B_1 + B_2$ . The decoder then cancels out  $B_1 + B_2$  and solves for  $A_1$  and  $A_2$  from the resulting two equations. The recovery is successful if and only if the resulting 2 equations are linearly independent. This amounts to the following condition:

$$\det \begin{pmatrix} \alpha_1 \beta_2 & \alpha_2 \beta_1 \\ \gamma_1 \delta_2 & \gamma_2 \delta_1 \end{pmatrix} = \alpha_1 \beta_2 \gamma_2 \delta_1 - \alpha_2 \beta_1 \gamma_1 \delta_2 \neq 0. \quad (2)$$

The left hand side of (2) is seen to be another multivariate polynomial in  $\xi$ ; denote this by  $Q_1(\xi)$ .

If a non-systematic part (e.g., node 4) fails, then we will perform a change of variables as in Figure 4:  $a_i = \gamma_i A_i + \delta_i B_i$ , for  $i = 1, 2$ . After the change of variables, the resulting code will be represented in terms of  $a_1, a_2, B_1, B_2$  and each coding coefficient will be a multivariate polynomial in  $\xi$  divided by another multivariate polynomial in  $\xi$ , or a rational function in  $\xi$ . Thus the corresponding condition for successful interference alignment has the form  $Q_i(\xi) \neq 0$ , where  $Q_i(\xi)$  is a multivariate polynomial in  $\xi$ . Here  $Q_i(\xi) \neq 0$  is the condition associated with the recovery of node  $i$ . Specifically, the condition for the recovery of node 4 is:

$$\det \begin{pmatrix} \frac{\delta_2}{\gamma_1 \gamma_2} & \frac{\delta_1}{\gamma_1 \gamma_2} \\ (\beta_2 - \frac{\alpha_2 \delta_2}{\gamma_2}) \frac{\alpha_1}{\gamma_1} & (\beta_1 - \frac{\alpha_1 \delta_1}{\gamma_1}) \frac{\alpha_2}{\gamma_2} \end{pmatrix} \neq 0, \quad (3)$$

which is equivalent to:

$$Q_4(\xi) = \gamma_1 \gamma_2 (\alpha_2 \delta_2 (\gamma_1 \beta_1 - \alpha_1 \delta_1) - \alpha_1 \delta_1 (\gamma_2 \beta_2 - \alpha_2 \delta_2)) \neq 0.$$

Integrating all these conditions together, the condition for being an MDS code and for successful interference alignment has the form  $Q(\xi) \neq 0$ , where  $Q(\xi) = P(\xi) Q_1(\xi) \dots Q_4(\xi)$  is a multivariate polynomial in  $\xi$ .

It is easy to verify that  $Q(\xi)$  is a non-zero polynomial. Applying the Schwartz–Zippel Lemma (Lemma 1), we see that for a sufficiently large finite field  $\mathbb{F}$ , when the coefficients in  $\xi$  are drawn i.i.d. and uniformly from  $\mathbb{F}$ ,  $\Pr[Q(\xi) \neq 0]$  can

be made arbitrarily close to 1. This in particular implies that for some finite field  $\mathbb{F}$ , there exists an assignment of  $\xi$  such that  $Q(\xi)$  evaluates to a non-zero number in  $\mathbb{F}$ .

**Lemma 1 (Schwartz–Zippel Lemma (see, e.g., [3])):**

Let  $Q(x_1, \dots, x_n) \in \mathbb{F}[x_1, \dots, x_n]$  be a multivariate polynomial of total degree  $d_0$  (the total degree is the maximum degree of the additive terms and the degree of a term is the sum of exponents of the variables). Let  $r_1, \dots, r_n$  be chosen independently and uniformly at random from  $\mathbb{F}$ . Then if  $Q(x_1, \dots, x_n)$  is a non-zero polynomial,  $\Pr[Q(r_1, \dots, r_n) = 0] \leq \frac{d_0}{|\mathbb{F}|}$ .

We want to point out that the above code existence proof follows a similar structure as the proof for the existence of capacity-achieving network codes for information multicasting by Koetter and Médard [4]. Specifically, in both contexts, the existence proofs are established by first formulating the existence condition as a product of polynomials, then showing each polynomial is nonzero, and finally applying the Schwartz–Zippel Lemma.

### B. Generalization to $(n, k)$ -MDS

We now generalize the scheme to the  $(n, k)$ -MDS case. Assume there are  $kq$  original information blocks of equal size, where  $q$  is a design parameter. Each storage node stores  $q$  blocks. Similar to Figure 1, the MDS code is formed by “stacking”  $q$   $(n, k)$ -MDS codes together. Let the first  $k$  nodes store the systematic parts. Let  $X_{1i}, \dots, X_{qi}$  denote the  $q$  blocks stored at node  $i$ , for  $i = 1, \dots, k$ . A non-systematic node, say  $j \in \{k+1, \dots, n\}$ , stores:

$$X_{m1}\xi_{m1}^{(j)} + \dots + X_{mk}\xi_{mk}^{(j)}, \quad \text{for } m = 1, \dots, q. \quad (4)$$

where  $\xi_{mi}^{(j)}$ ,  $m = 1, \dots, q$ ,  $i = 1, \dots, k$  are  $qk$  coding coefficients at node  $j$ . Let  $\xi$  denote the set of  $qk(n-k)$  coding coefficients  $\{\xi_{mi}^{(j)}\}$  in the nonsystematic parts.

Suppose a systematic node, say node 1, fails. We let node 2 pick one alignment direction, e.g.,  $X_{12} + \dots + X_{q2}$ . Nodes  $3, \dots, k$  can provide up to  $(k-2)q$  blocks. Each non-systematic node can produce one combination block whose interference space is aligned to the given direction  $X_{12} + \dots + X_{q2}$ . Thus a total of  $1 + (k-2)q + (n-k)$  blocks can be provided. The basic interference alignment approach aligns  $q$  unknowns into one direction, so that  $q$  unknowns are effectively replaced by one unknown. Therefore, we need  $(k-1)q + 1$  equations for decoding. For decoding to be successful, we require  $1 + (k-2)q + (n-k) \geq (k-1)q + 1$ , which reduces to the condition  $q \leq n - k$ .

There can be multiple ways for providing the needed  $(k-1)q + 1$  equations. For example, we can recover node 1 by downloading 1 block from node 2,  $q(k-2)$  blocks from nodes  $3, \dots, k$ , and 1 block each from nodes  $k+1, \dots, k+q$ . The proof of the following Lemma 2 assumes this method is used.

Suppose a nonsystematic node, say node  $n$ , fails. We introduce new variables  $X'_{11}, \dots, X'_{q1}$  as in Figure 4:

$$X'_{m1} \triangleq X_{m1}\xi_{m1}^{(n)} + \dots + X_{mk}\xi_{mk}^{(n)}, \quad m = 1, \dots, q \quad (5)$$

Then we treat  $\{X'_{m1}, X_{m2}, \dots, X_{mk}\}_{m=1}^q$  as the set of original information blocks. After this variable change, the resulting

situation becomes similar to the repair of a systematic node. Specifically, node 1 stores:

$$X_{m1} = (X'_{m1} - X_{m2}\xi_{m2}^{(n)} - \dots - X_{mk}\xi_{mk}^{(n)})/\xi_{m1}^{(n)}, \quad (6)$$

for  $m = 1, \dots, q$ . For  $j \in \{k+1, \dots, n-1\}$ , node  $j$  stores:

$$X'_{m1} \frac{\xi_{m1}^{(j)}}{\xi_{m1}^{(n)}} + \sum_{h=2}^k X_{mh} \left( \xi_{mh}^{(j)} - \frac{\xi_{mh}^{(n)} \xi_{m1}^{(j)}}{\xi_{m1}^{(n)}} \right) \quad (7)$$

for  $m = 1, \dots, q$ . We can recover node  $n$  by downloading 1 block from node 1, 1 block  $X_{12} + \dots + X_{q2}$  from node 2,  $q(k-2)$  blocks from nodes  $3, \dots, k$ , and 1 block each from nodes  $k+1, \dots, k+q-1$ .

As in Section II-A, we can establish a condition for the resulting code to be an MDS code and for the interference alignment technique to work. The condition is of the form

$$Q(\xi) = P(\xi)Q_1(\xi)Q_2(\xi) \dots Q_n(\xi) \neq 0, \quad (8)$$

where each term in the product is a multivariate polynomial with variables  $\xi$ . The term  $P(\xi)$  arises from the condition that the code is an  $(n, k)$ -MDS code. The term  $Q_i(\xi)$  arises from the condition that the interference alignment technique can be successfully applied to reduce the traffic for repairing node  $i$ .

**Lemma 2:**  $Q(\xi) = P(\xi)Q_1(\xi)Q_2(\xi) \dots Q_n(\xi)$  is a non-zero polynomial.

**Proof:** We show that each term in the product is a non-zero polynomial, for which we just need to show that it evaluates to a nonzero value for a certain assignment of the values in  $\xi$ .

To show  $P(\xi) \neq 0$ , note that we can set the coefficients of each row based on any given systematic MDS code.

To show  $Q_1(\xi) \neq 0$  for some assignment of  $\xi$ , note that we have all the freedom in setting  $\xi$ . To recover  $X_{11}, \dots, X_{q1}$ , we can assign the coefficients such that each block  $X_{i1}$  is provided directly by node  $k+i$  as one of its stored blocks. In Figure 3, we can set  $\alpha_1 = 1$ ,  $\gamma_2 = 1$ , and the other variables to 0. Then  $A_1$  is provided by node 3 and  $A_2$  is provided by node 4. Similarly, we can show  $Q_i(\xi)$  is a nonzero polynomial for  $i = 2, \dots, k$ .

We now show  $Q_n(\xi) \neq 0$  for some assignment of  $\xi$ . Set  $\xi_{mi}^{(n)} = 1$  for all  $m$  and  $i$ . Set  $\xi_{mh}^{(j)} = 1 + \xi_{m1}^{(j)}$  for  $m \in \{1, \dots, q\}$ ,  $h \in \{2, \dots, k\}$ ,  $j \in \{k+1, \dots, n-1\}$ . Substituting these values into (6) and (7), we see that the successful recovery is equivalent to the condition that:

$$\det \begin{pmatrix} 1 & 1 & \dots & 1 \\ \xi_{11}^{(k+1)} & \xi_{21}^{(k+1)} & \dots & \xi_{q1}^{(k+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{11}^{(k+q-1)} & \xi_{21}^{(k+q-1)} & \dots & \xi_{q1}^{(k+q-1)} \end{pmatrix} \neq 0. \quad (9)$$

Since we have the freedom in setting all variables in the above condition, the above condition can be satisfied for some assignment of  $\xi$ . Similarly, we can show that  $Q_j(\xi) \neq 0$  for  $j \in \{k+1, \dots, n\}$ .

Putting these together, we have established that  $Q(\xi)$  is a non-zero polynomial. ■



**Theorem 1 (Existence and Code Construction):**

For any  $q \leq n - k$ , there exists a finite field  $\mathbb{F}$  and an assignment of  $\xi$  from  $\mathbb{F}^{|\xi|}$  such that  $Q(\xi) \neq 0$ , which implies:

- The resulting code is an  $(n, k)$ -MDS code.
- The interference alignment technique can be successfully applied to repair each node  $i \in \{1, \dots, n\}$  by downloading a total of  $(k - 1)q + 1$  blocks from  $k + q - 1$  nodes.

Furthermore, for sufficiently large finite field  $\mathbb{F}$ , when the coefficients in  $\xi$  are drawn i.i.d. and uniformly from  $\mathbb{F}$ ,  $\Pr[Q(\xi) \neq 0]$  can be made arbitrarily close to 1.

**Proof:** Lemma 2 shows that  $Q(\xi)$  is a non-zero polynomial. The total degree of  $Q(\xi)$  is a function of the size of the problem. The claims then follow from the Schwartz–Zippel Lemma. ■

Since each block has size  $B/(kq)$ , this basic interference alignment scheme reduces the exact repair bandwidth from  $B$  to  $\frac{B((k-1)q+1)}{kq}$ . Using the cut analysis of [2], we can obtain a lower bound on the total repair bandwidth, which is  $\frac{Bd}{k(d-k+1)}$  if the new node downloads from  $d$  nodes. Such a lower bound can be achieved for the functional repair problem, via network coding. For  $k = 2$ , by setting  $q = d - 1$ , we see that the lower bound can be achieved by the interference alignment scheme. However, whether such a lower bound is tight for the exact repair of general  $(n, k)$  codes remains open.

### III. GROUP INTERFERENCE ALIGNMENT

The basic interference alignment scheme of Section II reduces the repair bandwidth but the saving diminishes as  $k$  gets large. In this section we present a technique called “group interference alignment”, which may lead to a smaller repair bandwidth for large  $k$ . As an analogy, we can think of the basic interference alignment scheme of Section II as the scalar version and the group interference alignment scheme as the vector version.

We begin by explaining the technique on a concrete setup – a  $(6, 4)$ -MDS code. Then we discuss the general  $(n, k)$  setup.

#### A. $(6, 4)$ -MDS with Group Interference Alignment

As illustrated by Figure 5, there are 8 original information blocks,  $A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2$ . Let  $\xi$  denote the vector formed by the 12 code coefficients  $\alpha, \beta, \gamma, \delta, \mu, \nu, \rho, \sigma, \theta_1, \theta_2, \eta_1, \eta_2$ . The code is formed by stacking together 2  $(6, 4)$ -MDS codes. Similar to Section II-A, the MDS requirement can be stated as the condition  $P(\xi) \neq 0$ , where  $P(\xi)$  is a multivariate polynomial in  $\xi$  that is a product of 30 determinants. Since we can set the coefficients so that each row is a systematic  $(6, 4)$  MDS code, it follows that  $P(\xi)$  is a nonzero polynomial.

We partition the 4 systematic nodes into 2 groups, each containing 2 nodes. The first group contains the original information blocks  $A_1, A_2, B_1, B_2$  and the second group contains the original information blocks  $C_1, C_2, D_1, D_2$ .

As illustrated by Figure 5, suppose node 1 fails, the decoder downloads  $C_1 + C_2$  from node 3 and  $D_1 + D_2$  from node 4. Then each non-systematic node generates a linear combination

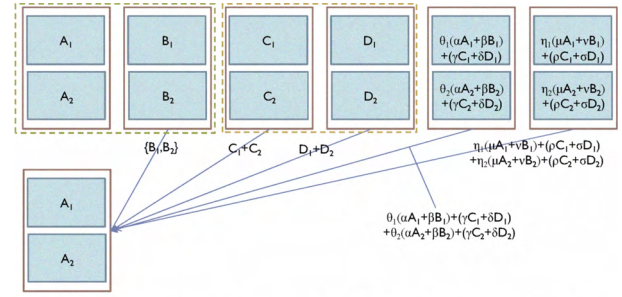


Fig. 5. Illustration of the group interference alignment technique.

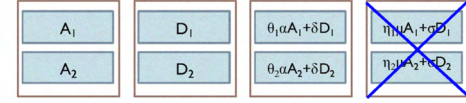


Fig. 6. The condition for recovering from a nonsystematic failure in Figure 5 is similar to that for a  $(4, 2)$  MDS illustrated here.

block of the form  $[A_1, A_2, B_1, B_2, C_1 + C_2, D_1 + D_2]v$ . In doing so, the interference signals  $C_1, C_2$  are aligned into a single dimension  $C_1 + C_2$  and the interference signals  $D_1, D_2$  are aligned into a single dimension  $D_1 + D_2$ , at the same time.

Note that to ensure the simultaneous alignment of  $C_1 + C_2$  and  $D_1 + D_2$ , the code needs to have some specific structure. In Figure 5, node 5 stores  $\theta_1\alpha A_1 + \theta_1\beta B_1 + \gamma C_1 + \delta D_1$  and  $\theta_2\alpha A_2 + \theta_2\beta B_2 + \gamma C_2 + \delta D_2$ . The key property is that the coefficients before  $A_i$  (resp.  $C_i$ ) have the same ratio as the coefficients before  $B_i$  (resp.  $D_i$ ).

In Figure 5, to recover node 1, the decoder downloads  $B_1$  and  $B_2$  from node 2 (within the same group),  $C_1 + C_2$  from node 3 and  $D_1 + D_2$  from node 4, and 2 linear combinations from the non-systematic nodes. This gives 6 equations in terms of the 6 unknowns  $A_1, A_2, B_1, B_2, C_1 + C_2, D_1 + D_2$ . It can be verified from Figure 5 that the recovery condition can be stated as  $Q_1(\xi) = \mu\alpha(\eta_1\theta_2 - \eta_2\theta_1) \neq 0$ .

Similarly, any systematic node can be repaired while downloading 6 equations. The corresponding condition for successful group interference alignment is of the form  $Q_i(\xi) \neq 0$ , and it can be verified that each  $Q_i(\xi)$  is a nonzero polynomial.

The group interference alignment scheme does not seem to be applicable to the recovery of a failed non-systematic node. However, we can still apply the basic interference alignment scheme of Section II-B. To recover a non-systematic node, say, node 6, we download  $B_1, B_2$  from node 2 and  $C_1, C_2$  from node 3. Since now the decoder has  $B_1, B_2, C_1, C_2$ , the problem essentially reduces to the recovery from a nonsystematic failure in a  $(4, 2)$ -MDS code, as illustrated by Figure 6. Further note that we essentially have the same degree of freedom in assigning the coefficients as in Figure 4. Therefore, using a similar argument as in Lemma 2, we can show that the resulting polynomial  $Q_6(\xi)$  is a non-zero polynomial.

Putting these together, the MDS condition and the requirement that the interference alignment techniques can be successfully applied amount to requiring  $Q(\xi) = P(\xi)Q_1(\xi) \dots Q_6(\xi) \neq 0$  for some assignment of  $\xi$  in a certain finite field  $\mathbb{F}$ . From the above discussion, we know that  $Q(\xi)$  is a nonzero polynomial. Hence from the Schwartz–

Zippel Lemma, for a sufficiently large finite field  $\mathbb{F}$ , when the coefficients in  $\xi$  are drawn i.i.d. and uniformly from  $\mathbb{F}$ ,  $\Pr[Q(\xi) \neq 0]$  can be made arbitrarily close to 1.

### B. $(n, k)$ -MDS with Group Interference Alignment

As before, we assume (i) there are  $kq$  original blocks of equal size, (ii) each storage node stores  $q$  blocks, (iii) the code is formed by stacking together  $q$   $(n, k)$ -MDS code, and (iv) the first  $k$  nodes store the systematic parts.

We partition the  $k$  systematic nodes into 2 groups,  $X$  and  $Y$ , containing  $p$  and  $k - p$  nodes, respectively. Let  $X_{1i}, \dots, X_{qi}$  denote the  $q$  blocks stored at node  $i$ , for  $i = 1, \dots, p$ . Let  $Y_{1i}, \dots, Y_{qi}$  denote the  $q$  blocks stored at node  $i$ , for  $i = p+1, \dots, k$ . A non-systematic node, say  $j \in \{k+1, \dots, n\}$ , stores:

$$\theta_m^{(j)} \sum_{l=1}^p X_{ml} \phi_l^{(j)} + \sum_{l=p+1}^k Y_{ml} \psi_l^{(j)}, \text{ for } m = 1, \dots, q \quad (10)$$

where  $\theta_*^{(j)}, \phi_*^{(j)}, \psi_*^{(j)}$  are  $q + k$  coding coefficients at node  $j$ . Let  $\xi$  denote the set of  $(q + k)(n - k)$  coding coefficients that specify the code.

When a systematic node in one group fails, we pick a direction for each node in the other group and let the remaining nodes align to these directions. For example, suppose node 1 fails. The decoder can download up to  $(p - 1)q$  original blocks from nodes  $2, \dots, p$ . Each node  $l$  in  $\{p + 1, \dots, k\}$  generates  $Y_{1l} + \dots + Y_{ql}$ . For  $j \in \{k + 1, \dots, n\}$ , node  $j$  can generate a mixture block  $\sum_{l=1}^p (\theta_1^{(j)} X_{1l} + \dots + \theta_q^{(j)} X_{ql}) \phi_l^{(j)} + \sum_{l=p+1}^k (Y_{1l} + \dots + Y_{ql}) \psi_l^{(j)}$ . Thus, up to  $(p - 1)q + (k - p) + (n - k)$  blocks can be generated in this way.

Using the downloaded blocks, the decoder solves for  $pq + (k - p)$  unknowns:  $X_{il}$  for  $i \in \{1, \dots, q\}, l \in \{1, \dots, p\}$ , and  $(Y_{1l} + \dots + Y_{ql})$  for  $l = p + 1, \dots, k$ . Therefore,

$$(p - 1)q + (k - p) + (n - k) \geq pq + (k - p), \quad (11)$$

or equivalently,  $q \leq n - k$ .

There can be multiple ways for providing  $pq + (k - p)$  equations. For example, we can download  $(p - 1)q$  original blocks from nodes  $2, \dots, p$ ,  $k - p$  blocks from nodes  $p + 1, \dots, k$ , and  $q$  blocks from nodes  $k + 1, \dots, k + q$ . For ease of explanation, the proof of Theorem 2 assumes this method is used.

Similarly, when a systematic node in the second group fails, the scheme downloads a total of  $(k - p)q + p$  blocks from  $p + (k - p - 1) + q = k + q - 1$  nodes.

If a non-systematic node fails, we apply the basic interference alignment technique of Section II-B to recover the failed node by downloading  $(k - 1)q + 1$  blocks over the network.

#### Theorem 2 (Existence and Code Construction):

For any  $p \in \{1, \dots, k - 1\}$  and  $q \leq n - k$ , there exists a finite field  $\mathbb{F}$  and an assignment of  $\xi$  from  $\mathbb{F}^{|\xi|}$  such that

- The resulting code is an  $(n, k)$ -MDS code.
- The group interference alignment technique can be successfully applied to repair each systematic node  $i \in \{1, \dots, p\}$  by downloading a total of  $pq + (k - p)$  blocks from  $k + q - 1$  nodes.

- The group interference alignment technique can be successfully applied to repair each systematic node  $i \in \{p + 1, \dots, k\}$  by downloading a total of  $(k - p)q + p$  blocks from  $k + q - 1$  nodes.
- The basic interference alignment technique can be successfully applied to repair each non-systematic node  $i \in \{k + 1, \dots, n\}$  by downloading a total of  $(k - 1)q + 1$  blocks from  $k + q - 1$  nodes.

Furthermore, for sufficiently large finite field  $\mathbb{F}$ , when the coefficients in  $\xi$  are drawn i.i.d. and uniformly from  $\mathbb{F}$ , the probability that the above 4 conditions are met can be made arbitrarily close to 1.

**Proof:** As in Section II-B, we define a multivariate polynomial  $Q(\xi)$ , with the property that  $Q(\xi) \neq 0$  implies the 4 conditions given in the theorem. The multivariate polynomial  $Q(\xi)$  will be of the form  $Q(\xi) = P(\xi)Q_1(\xi) \dots Q_n(\xi)$ , where  $P(\xi)$  corresponds to the MDS condition, and  $Q_i(\xi)$  corresponds to the condition for the recovery of node  $i$ .

To show  $P(\xi) \neq 0$ , note that we can set the coefficients of each row based on any given systematic MDS code.

Now consider the condition for the recovery of node 1. It can be verified that the condition reduces to:

$$Q_1(\xi) = \det \begin{pmatrix} \theta_1^{(k+1)} \phi_1^{(k+1)} & \dots & \theta_q^{(k+1)} \phi_1^{(k+1)} \\ \vdots & & \vdots \\ \theta_1^{(k+q)} \phi_1^{(k+q)} & \dots & \theta_q^{(k+q)} \phi_1^{(k+q)} \end{pmatrix} \neq 0.$$

To show  $Q_1(\xi) \neq 0$ , we set  $\phi_* = 1$  and choose  $\{\theta_m^{(j)}\}$  so that the above matrix is invertible. Similarly, we can show that for  $i = 1, \dots, k$ , the condition for the successful recovery of node  $i$ ,  $Q_i(\xi) \neq 0$ , can be satisfied for some assignment of  $\xi$ .

Now consider the condition for the recovery of a non-systematic node, say, node  $n$ . To recover node  $n$ , we download  $(k - 2)q$  systematic blocks from nodes  $2, \dots, k - 1$ . Then as illustrated by Figure 6, the condition for the successful recovery of node  $n$  reduces to that of a  $(2 + (n - k), 2)$ -MDS code, where the two systematic nodes store  $X_{11}, \dots, X_{q1}$  and  $Y_{1k}, \dots, Y_{qk}$ , respectively, and a non-systematic node  $j$  stores  $\theta_m^{(j)} X_{m1} \phi_1^{(j)} + Y_{mk} \psi_k^{(j)}$  for  $m = 1, \dots, q$ . Then using a similar argument as in Lemma 2, we can show that  $Q_n(\xi) \neq 0$  for some assignment of  $\xi$ . Similarly, we can show  $Q_j(\xi)$  is a nonzero polynomial, for  $j \in \{k + 1, \dots, n\}$ .

Therefore  $Q(\xi)$  is a non-zero polynomial. The total degree of  $Q(\xi)$  is a function of the problem size. Applying the Schwartz-Zippel Lemma, the claim is established. ■

#### REFERENCES

- [1] V. R. Cadambe and S. A. Jafar, "Interference alignment and the degrees of freedom for the  $K$  user interference channel," *IEEE Transactions on Information Theory*, 54(8), pp. 3425–3441, Aug. 2008.
- [2] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *Submitted for journal publication. Preliminary version appeared in Infocom 2007*.
- [3] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.
- [4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, Oct. 2003.
- [5] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," In *Allerton Conference on Control, Computing, and Communication*, Urbana-Champaign, IL, September 2007.